



VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ

BRNO UNIVERSITY OF TECHNOLOGY

FAKULTA INFORMAČNÍCH TECHNOLOGIÍ

FACULTY OF INFORMATION TECHNOLOGY

ÚSTAV POČÍTAČOVÝCH SYSTÉMŮ

DEPARTMENT OF COMPUTER SYSTEMS

FILMOTÉKA

FILM LIBRARY

BAKALÁŘSKÁ PRÁCE

BACHELOR'S THESIS

AUTOR PRÁCE

AUTHOR

VLADYSLAV NIMIZHAN

VEDOUCÍ PRÁCE

SUPERVISOR

Ing. MARTIN KRČMA

BRNO 2017

Vysoké učení technické v Brně - Fakulta informačních technologií

Ústav počítačových systémů

Akademický rok 2016/2017

Zadání bakalářské práce

Řešitel: **Nimizhan Vladyslav**

Obor: Informační technologie

Téma: **Filmotéka**
Film Library

Kategorie: Web

Pokyny:

1. Nastudujte možnosti získávání informací o filmech a seriálech online v internetových databázích. Prostudujte rozhraní pro práci s těmito databázemi.
2. Navrhněte systém, který umožní uživateli vést databázi zhlédnutých filmů a seriálů, aktuálně sledovaných seriálů, včetně jejich hodnocení a poznámek (filmotéku) a dále umožní plánovat další sledování. Na naplánované sledování bude systém umět uživatele vhodnou formou upozorňovat. K vloženým záznamům bude systém automaticky hledat dodatečné informace ve vybraných databázích a integrovat je s informacemi vloženými uživatelem. Systém bude schopen generovat různé statistiky o sledování. Dále bude systém vybaven moderním vícejazyčným uživatelským rozhraním.
3. Navržený systém implementujte jako webovou službu.
4. Získejte a vyhodnoťte zpětnou vazbu od skupiny nezávislých uživatelů, testerů aplikace.
5. Navrhněte směr další práce.

Literatura:

- dle pokynů vedoucího

Pro udělení zápočtu za první semestr je požadováno:

- Splnění bodů 1 a 2.

Podrobné závazné pokyny pro vypracování bakalářské práce naleznete na adrese

<http://www.fit.vutbr.cz/info/szz/>

Technická zpráva bakalářské práce musí obsahovat formulaci cíle, charakteristiku současného stavu, teoretická a odborná východiska řešených problémů a specifikaci etap (20 až 30% celkového rozsahu technické zprávy).

Student odevzdá v jednom výtisku technickou zprávu a v elektronické podobě zdrojový text technické zprávy, úplnou programovou dokumentaci a zdrojové texty programů. Informace v elektronické podobě budou uloženy na standardním nepřepisovatelném paměťovém médiu (CD-R, DVD-R, apod.), které bude vloženo do písemné zprávy tak, aby nemohlo dojít k jeho ztrátě při běžné manipulaci.

Vedoucí: **Krčma Martin, Ing., UPSY FIT VUT**

Datum zadání: 1. listopadu 2016

Datum odevzdání: 17. května 2017

VYSOKÉ UČENÍ TECHNICKÉ V BRNĚ
Fakulta informačních technologií
Ústav počítačových systémů a sítí
602 00 Brno, Božetěchova 2

L.S.



prof. Ing. Lukáš Sekanina, Ph.D.
vedoucí ústavu

Abstrakt

Tato bakalářská práce se zabývá vedením filmotéky. Uživatelé mají možnost vést databázi zhlednutých filmů a seriálů, včetně jejich hodnocení a poznámek, vkládat oblíbené filmy a seriály do vlastního seznamu. Další možnosti je přidávání vysílaných epizod seriálu do kalendáře. Na naplánované sledování uživatel dostává upozornění na zařízení které je synchronizováno s kalendářem. Při tvorbě aplikaci byl důležitý návrh samotného uživatelského rozhraní a intuitivnost systému. Výsledkem práce je aplikace, implementovaná v jazyce PHP 5.6 s využitím frameworku Nette.

Abstract

This bachelor thesis is devoted to run film library. Users can manage databse of seen films and serials including ratings and notes. They can also add runntime of oncoming episodes to calendar. User gets notifications about planned episodes on their device synchronised with calendar. By designing of the aplication user interface and and intuitive interface was very important. Result of this work is an application implemented in PHP 5.6 with use of framework Nette.

Klíčová slova

web, databáze, PHP, Nette, filmotéka, HTML, CSS, API.

Keywords

web, database, PHP, Nette, film library, HTML, CSS, API.

Citace

NIMIZHAN, Vladyslav. *Filmotéka*. Brno, 2017. Bakalářská práce. Vysoké učení technické v Brně, Fakulta informačních technologií. Vedoucí práce Ing. Martin Krčma

Filmothéka

Prohlášení

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně pod vedením pana Ing. Martina Krčmy. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

.....

Vladyslav Nimizhan

17. května 2017

Poděkování

Rád bych poděkoval Ing. Martinovi Krčmovi, za vedení bakalářské práce, odbornou pomoc a přátelský přístup.

Obsah

1	Úvod	3
2	Teoretická část	4
2.1	Existující řešení	4
2.1.1	Filmová databáze IMDb	4
2.1.2	Seriálová databáze TVmaze	5
2.1.3	TheTVDB	6
2.1.4	Episode Calendar	7
2.1.5	Česko-Slovenská filmová databáze	8
2.2	Shrnutí	9
3	Návrh aplikace	10
3.1	Klíčové funkce a požadavky systému	10
3.2	Návrh uživatelského rozhraní	11
3.2.1	Domácí stránka	11
3.2.2	Zobrazení filmů a seriálů	12
3.3	Diagram případů užití	13
3.4	Návrh databáze	13
3.5	Architektura webové aplikace	14
3.5.1	Použité technologie	15
3.5.2	Framework	16
3.6	Vybrané API	17
3.6.1	AddEvent	17
3.6.2	IMDb	17
3.6.3	TVMaze	17
4	Implementace	19
4.1	Serverová část	19
4.1.1	Vyhledání	19
4.1.2	Parsování	20
4.1.3	Přidání do kalendáře	21
4.2	Databazová část	23
4.2.1	Připojení k databázi MySQL	23
4.2.2	Přihlašování uživatelů	24
4.2.3	Přidávání do watchlistu	25
4.2.4	Statistiky	25
5	Testování	27

5.1	Návrh testovacího formuláře	27
5.2	Finální testování	28
6	Závěr	33
	Literatura	34
	Přílohy	36
A	Obsah CD	37

Kapitola 1

Úvod

V době, kdy jediná forma ukládání filmu byla prostřednictvím zachycení obrazového a zvukového záznamu na pozitivní filmové kopii (zpravidla o šířce 8, 16, 35 nebo 70 mm), odpovědnosti filmoték byla systematizace, restaurování a uschování díla. S příchodem a rozšířením elektronických nosičů filmotéky začaly převádět svoje filmové archívy do digitální podoby která umožňuje mnohem jednodušší techniku kopírování a rozšíření filmů.

Cílem bakalářské práce je vedení filmotéky, kde uživatele budou moct sledovat filmy a seriály zároveň, uchovávat historii zhlednutí, získávání informací o filmech a seriálech, sdílet oblíbené seriály a dostávat upozornění nových epizodu sledovaných seriálů. Motivaci k tomuto tématu slouží největší filmová databáze IMDb [6] a autorem dlouhodobě využívaná seriálová databáze TVmaze [10]. Dalším cílem je kvalitní návrh a konkurence těmto projektům z hlediska grafické prezentace, absenci reklamy, jednoduchosti a srozumitelnosti uživatelského rozhraní.

Práce je rozdělena do několika okruhů. V tomto dokumentu se nejdříve seznámíme s existujícími řešeními. Základem každé aplikace je návrh na který se zaměřuje další kapitola. Tato část popisuje výběr použitých technologií pomocí kterých aplikace je vyvíjena. Závěr této kapitoly se zaměřuje na API, které bylo použito k získání podrobných informací o jednotlivých filmech a seriálech. Poslední kapitola se věnuje implementaci a popisuje jakým způsobem byly implementovány důležité části programu. Na konci dokumentu najdeme závěrečné shrnutí, výsledky testování a další možný vývoj.

Kapitola 2

Teoretická část

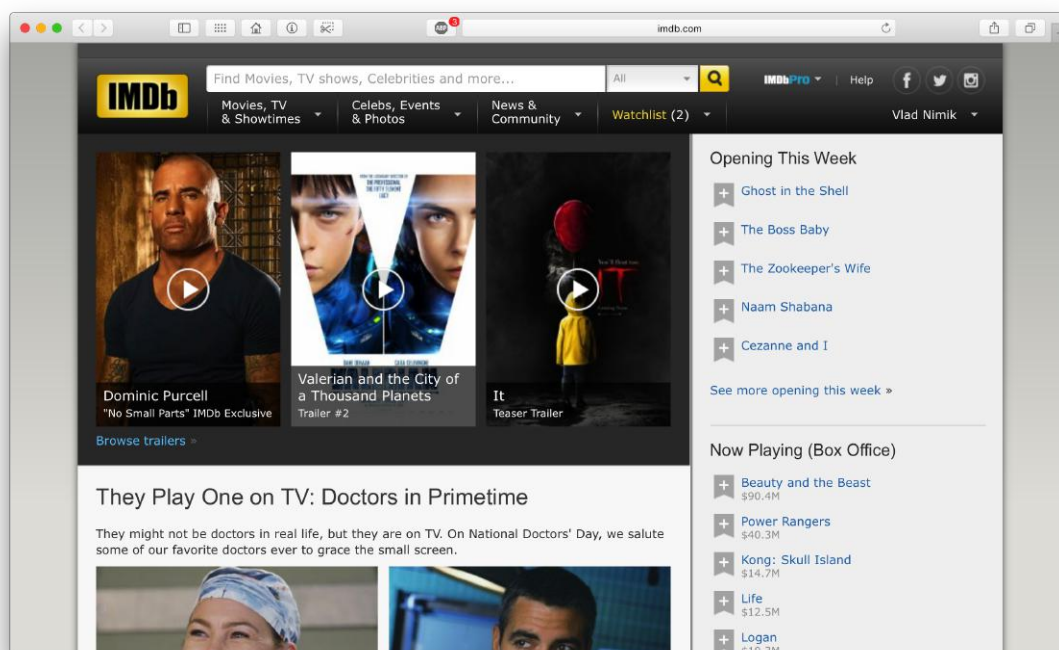
Při získávání podkladů byla studována již existující řešení webových filmoték, které budou popsány na začátku kapitoly. Dál popíšeme funkcionalitu programu, jejich výhody a nevýhody, které ovlivní návrh výsledné aplikace. Následně se popíšu funkce které inspirovaly vývoj programu. Závěr kapitoly bude obsahovat shrnutí uvedených poznatků.

2.1 Existující řešení

Aplikace pro vedení vlastní filmotéky není novinkou a proto na internetu existuje několik kvalitních řešení. Při hledání podobných řešení byly nalezeny jenom některé aplikace které splňovaly pouze malou část uvedených požadavků. Populární aplikace se liší v různých funkcích a cenou na jejich použití. Tato kapitola analyzuje existující řešení, cenu poskytovaných funkcí, design a první dojmy při používání aplikace.

2.1.1 Filmová databáze IMDb

Internet Movie Database je webová stránka a světově největší databáze informací o více než 3.9 miliónů filmů, televizních seriálu, 7.5 milionů osobností spojených s filmovou tvorbou. Databáze poskytuje informace o aktuálním žebříčku filmů, které jsou nejnovější a nejsledovanější. U každého z filmů uživatel může jediným kliknutím zjistit, kdo v něm hraje, podívat se na upoutávku a přechíst si nejzajímavější informace. Jednou z populárních možností *IMDb* je online hodnocení. Registrované uživatelé mají možnost hodnotit filmy. Na stupnici od 1 (nejhorší) do 10 (nejlepší) může každý z dlouhodobě registrovaných uživatelů vyjádřit své sympatie k příslušnému filmovému titulu. Na základě hodnocení uživatelů je sestaven žebříček 250 nejlepších filmu všech dob, a taky 100 nejhorších filmu. V tomto žebříčku TOP-250 se neberou v úvahu hodnocení všech hodnotících. Započítané jsou pouze hlasy trvalých uživatelů. Tímto způsobem je možné předcházet podvádění přičítání hlasů. Nevýhodou *IMDb* je naplnění webové stránky která obsahuje hodně nepotřebných informací včetně reklamy ve které uživatel se ztrácí. Další nevýhodou je absence kalendáře pro jednotlivé epizody. [6]

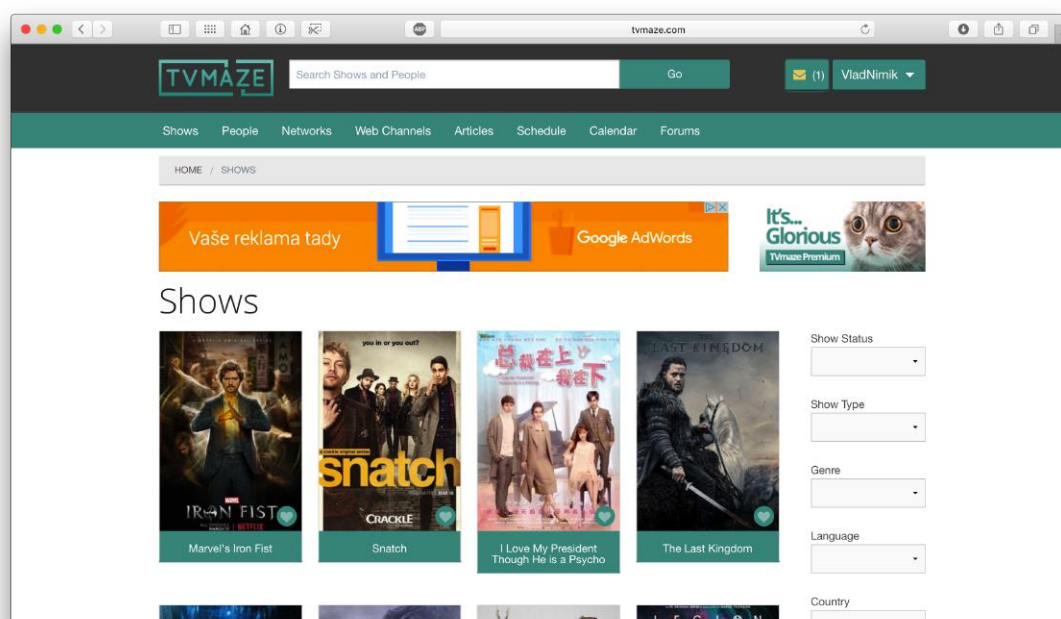


Obrázek 2.1: Filmová databáze IMDb

2.1.2 Seriálová databáze TVmaze

TVmaze je komunita televizních a specializovaných přispěvatelů. Každodenně se databáze aktualizuje a poskytuje uživatelům informací o seriálech, hercích, epizodách a jejich podrobnostech. Další možnosti je sledování dnešních televizních pořadů a následujících

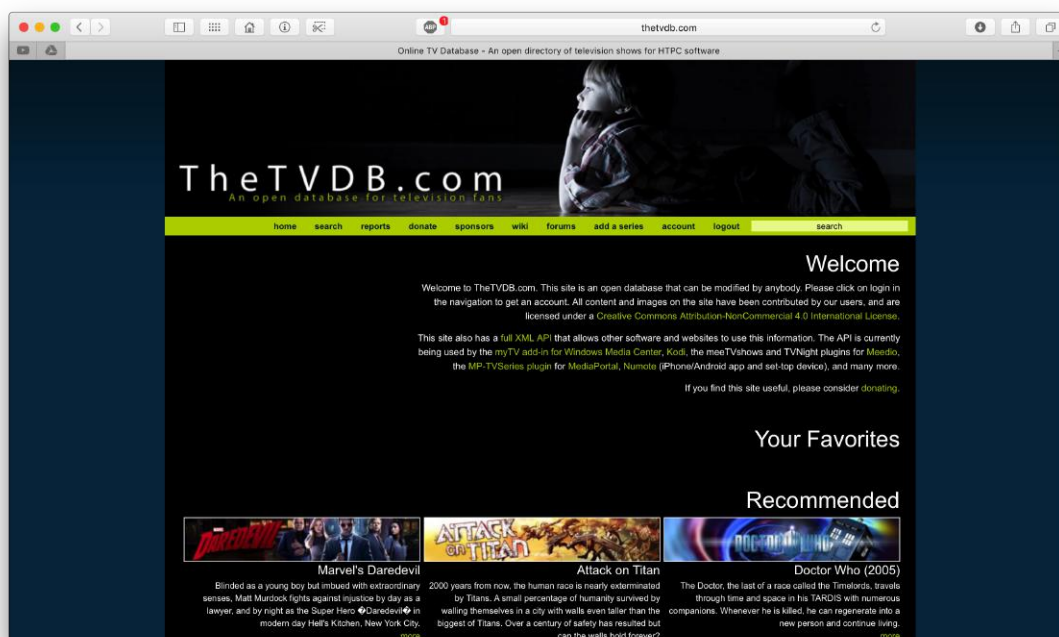
epizod. Uživatel má možnost vyhledat pořad podle typu, stavu, žánru a jazyků. Stejně jako u *IMDb* stránka poskytuje možnost hodnocení seriálu trvalým uživatelům, podle kterého sestavuje zebříček nejlepších seriálů a programu. Na rozdíl od *IMDb* stránka poskytuje moderní design a omezený funkcionalitou kalendář. Pro přidání nových a rozšíření původních funkcí aplikace nabízí premium účty v ceně od 36 do 120 € ročně. [10]



Obrázek 2.2: Seriálová databáze TVMaze

2.1.3 TheTVDB

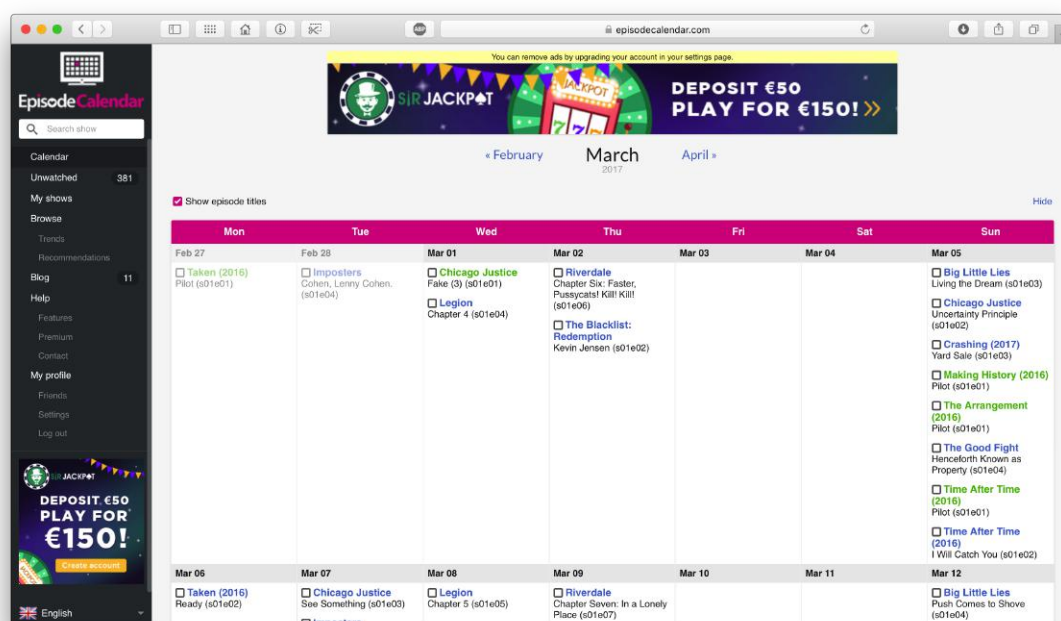
TheTVDB - otevřená databáze televizních seriálů v různých jazycích co je velkou výhodou oproti jiným databázím. Webová stránka obsahuje informace, kterou uživatelé můžou přidávat nebo měnit. Veškeré informace a obrázky na stránkách jsou nahrány uživateli pro ostatní uživatele. Databáze a webová stránka jsou otevřené pro používání a šíření *GNU GPL* licenci[2]. Web používá moderovanou editaci pro udržování vlastní normy a zlepšení kvality poskytovaných informací. Pro vývojáře databáze nabízí *JSON API*. Nevýhodou aplikace je nemoderní design, chybějící informace o následujících epizodách a jejich možnost přidávání do kalendáře. Uživatel tedy neobdrží žádné upozornění o nastávajících epizodách [1].



Obrázek 2.3: Seriálová databáze TheTVDB

2.1.4 Episode Calendar

Episode Calendar - další aplikace televizních pořadů, která získává data z databáze TheTVDB. Kvůli nedostatku informací o seriálu a jednotlivých epizodách lze říci, že web je více zaměřen na práci s kalendářem, který je zobrazen hned na hlavní stránce. Základní verze aplikace obsahuje žebříček nejlepších seriálů, ukazuje pokrok profilu vůči zhlédnutým epizodám, umožňuje přidávat epizody do “watchlistu” a importovat je do servisu Kalendáře Google. V placené verzi jsou zahrnovány “achievements”, absence reklamy, možnost vytvoření vlastního seznamu a možnost nahlédnutí do kalendářů přátel. V placené verzi má uživatel také možnost zrušení omezeného počtu sledovaných seriálů. Maximální počet, které si uživatel může přidat do svého kalendáře, je v neplacené verzi omezen na 20 seriálů [3].



Obrázek 2.4: Seriálová databáze Episode Calendar

2.1.5 Česko-Slovenská filmová databáze

Česko-Slovenská filmová databáze (*dále jako ČSFD*) je databáze filmů, televizních pořadů a sociální síť pro filmové fanoušky. *ČSFD* obsahuje téměř 502 tisíc filmů a 350tisíc registrovaných uživatelů. Hodnocení a recenze filmů – jsou veřejné a z obsáhlé databáze filmů tvoří mapu jejich kvality. *ČSFD* je českou obdobou databáze *IMDb*. Na rozdíl od ní umožňuje registrovaným uživatelům vést si přehled vlastní filmotéky, nabízí program všech českých a slovenských kin, přehled filmů vycházejících v kinech, nebo taky program českých a slovenských televizních stanic. Nedílnou součástí tohoto webu jsou statistiky a žebříčky, např. 300 nejoblíbenějších, nejlepších, nejpopulárnějších nebo nejhůře hodnocených filmů a seriálů.

ČSFD má taky svoje nevýhody. Jednou z nich je absence jiných jazyků. Web je tedy určen jenom pro české a slovenské uživatele. Další nevýhodou je neumožnění sledování nových epizod u seriálu, protože databáze neposkytuje žádný kalendář. Další nevýhodou (negativem) je spousta zbytečných informací, které zatěžují webovou stránku a uživatele např. žhavé online filmy, DVD tipy (v dnešní době lidé dávají přednost online službám např. Netflix před kupováním DVD), nejnavštěvovanější videa a nejoblíbenější uživatelé dle krajů [11].



Obrázek 2.5: Česko-Slovenská filmová databáze

2.2 Shrnutí

Existuje několik aplikací zaměřených na sledování seriálů ze kterých si uživatel může vybrat nejlepší podle vlastních požadavků. První nevýhodou podobných aplikací je to, že jsou funkčně omezené a pro využití rozšířených funkcí, uživatel musí kupovat premium verzi programu a pravidelně platit každý měsíc nebo rok v závislosti od vybrané aplikace. Další nevýhodou je, že uživatel nemá k dispozici zkušební verzi premium účtu. Je tedy nucen předplatit si danou službu bez možnosti její vyzkoušení. Tato vlastnost odradí většinu uživatelů, který tak často sáhnou po jiné internetové službě. Následující nevýhodou existujících programů je nemoderní vzhled, častý výskyt reklamy a velmi omezená přizpůsobivost kvůli velkému počtu nadbytečných informací. Z aplikací uvedených výše umožňuje souběžné sledování filmů a seriálů jenom IMDb. Tak jako každá jiná aplikace IMDb taky má své nevýhody. IMDb neposkytuje uživateli možnost synchronizace informací se svým kalendářem. Není teda možné přidávat sledované seriály resp. filmy do svého denního rozvrhu a dostávat patřičné upozornění. Dalším nedostatkem je nepřítomnost české lokalizace, pouze TheTVDB poskytuje informace v různých jazycích, která většinou není úplná kvůli malému zájmu a použití ze strany českých uživatelů, které by měli sami přidávat a starat se o informací na webové stránce.

Na základě provedené analýzy bylo zjištěno, že dostupnou aplikaci, která by splňovala všechny potřebné požadavky na internetu není. Podobné aplikace jsou placené nebo nemoderní s nedostačující přizpůsobivostí.

Kapitola 3

Návrh aplikace

Tato kapitola je zaměřena na návrh řešení. Jako první rozebereme klíčové funkce aplikace a obecně požadavky na informační systém. Pokračování návrhu proběhne zaměřením na grafické modely a jejich jednotlivé části. Dále bude na základě specifikace a analýzy požadavků vytvořen diagram případů užití (Use case diagram) a datový model potřebný pro vytvoření aplikace. Závěr kapitoly se zabývá vybranými API, dostupnými technologiemi a následně jejich analýzou a výběrem konkrétně technologií pro vývoj.

3.1 Klíčové funkce a požadavky systému

Každá filmotéka nabízí základní množství vlastností a funkcí. Je zřejmé, že jednotlivé funkce a vlastnosti se mohou v závislosti od konkrétního zaměření dané filmotéky značně lišit. Základním požadavkem pro takový informační systém jsou obecně informace o filmech a seriálech. Tento základ může být obalen dalšími pokročilými informacemi jako hodnocení uživatelů, popis jednotlivých epizod a sezón, datum zveřejnění seriálu a datum další epizody.

Pokud je seriál právě vysílaný, uživatel má k dispozici odkaz na jeho následující epizodu. Odkaz obsahuje popis epizody, čas zveřejnění podle časového pásma a funkce přidání epizody do vlastního kalendáře. Tím uživatel obdrží včas upozornění a nezapomene na žádnou další epizodu jeho oblíbeného pořadu.

Další možností je, že jednotlivý seriál obsahuje všechny data epizod. Pokud datum epizody je v blízké budoucnosti, epizody se budou zobrazovat v sekci nad ostatními epizodami. Tím pádem si uživatel může přidat všechny další epizody do svého kalendáře. Tato funkce zachraňuje uživatele před stálým přidáváním jednotlivých epizod a tím pádem eliminuje riziko zapomenutí na další epizodu.

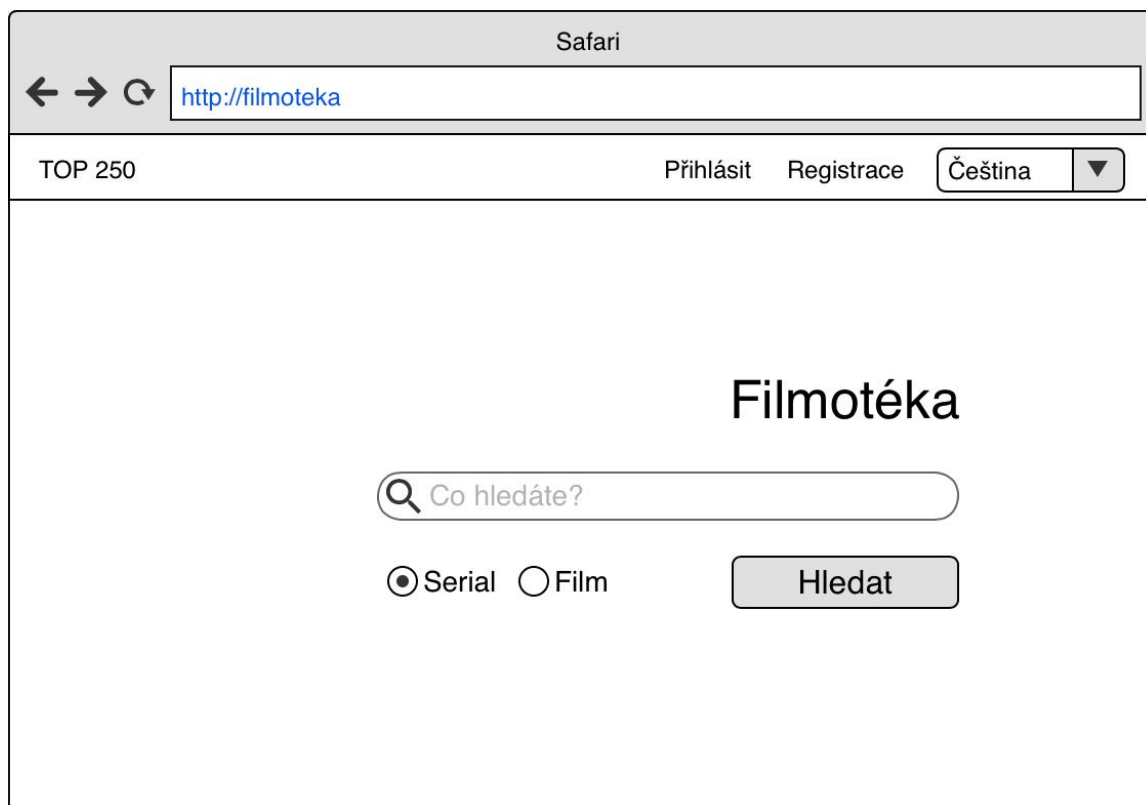
Registrování uživatele mají možnost ohodnotit seriál nebo film prostřednictvím komentářů. Hodnotit tak mohou filmy a seriály ještě před samotným přidáním seriálu a filmu do jejich watchlistu. Každý seriál bude obsahovat jednotlivé hodnocení podle filmové databáze *IMDb* a registrovaných uživatelů naší aplikace. Kromě toho taky bude obsahovat zvlášť pozitivní a negativní komentáře na základě získaných bodů od uživatelů. Popularita seriálu stoupá na základě počtu lidí, které tento seriál sledují a je zobrazená u jednotlivých filmu nebo seriálu.

3.2 Návrh uživatelského rozhraní

Tato sekce se věnuje návrhu GUI aplikace. Hlavním cílem je určit jakým způsobem budou data vizuálně zobrazována uživateli. Základem bylo vytvořit jednoduché prostředí přístupné pro všechny s přehledným a responzivním designem. Taktéž si navrhne z jakých částí se prostředí bude skládat a kam umístíme jednotlivé prvky. Pro lepší vysvětlení jsou uvedené grafické schémata které byly vytvořené pomocí nástroje Moqups [7].

3.2.1 Domácí stránka

Nejdřív bylo nutné navrhnout hlavní vzhled stránky, kde se bude nacházet formulář na vyhledávání filmu nebo seriálu. Ten bude obsahovat tlačítko, kterým dojde k odeslání formuláře. Tento způsob zobrazení domácí stránky je zvolen z důvodů jednoduššího vyhledání potřebných informací. Vzhled domácí stránky lze vidět na obrázku ???. Webová stránka je rozdělena na dvě části a to na navigační a zobrazovací část



Obrázek 3.1: Návrh domácí stránky

Navigační část se bude zobrazovat v horní části každé stránky jako horizontální pruh. V levé části bude obsahovat odkaz na seznam nejlepších seriálu. V pravé části se nachází odkaz na registraci a přihlašování uživatele. Dalším prvkem je rozbalovací menu s možností výběru jazyků aplikace.

Zobrazovací část na rozdíl od navigační části mění svůj obsah v závislosti od navštívené stránky.

3.2.2 Zobrazení filmů a seriálů

Jednou z hlavních činností, kterou uživatel bude často provádět je přidávání nových seriálů do watchlistu. K tomu slouží samostatná stránka ke které se uživatel dostane skrz vyhledávání a výběr potřebného seriálu ze seznamu všech seriálů s podobným nebo stejným názvem. Obsah stránky se mění podle toho, jestli uživatel je nebo není přihlášen. Stránka je rozdělena na 2 části. První část je horizontální seznam který se skládá z odkazu na:

Sezóny které obsahují plakát, popis a další údaje o jednotlivých sezónách seriálu: úvodní část, poslední část, počet epizod, kanál a země.

Epizody obsahuje seznam všech epizod seriálu včetně popisu a plakátu.

Další Epizoda poskytuje aktuální informaci o další epizodě a možnost přidání epizody do kalendáře uživatele.

Ukázka obsahuje odkaz na trailer daného seriálu.

Druhá část je sekce, která obsahuje plakát, popis seriálu a tabulku s různými údaji o seriálu, která zahrnuje informaci o jazyku, žánru, délce epizody, datu premiéry, statusu, hodnocení registrovaných uživatelů a fanoušků z webové aplikace IMDb. Registrovaným uživatelům se navíc zobrazují komentáře jiných uživatelů a taktéž je jim poskytnuta možnost ohodnotit nebo okomentovat seriál a přidat ho do svého seznamu sledovaných seriálů.

Safari

← → ↻

http://filMOTEKA

TOP 250

Přihlásit Registrace Čeština ▼

SezónyEpizodyDalší EpizodaUkázka



Scherlock

Sherlock Holmes and Dr John Watson's adventures in 21st Century London. A thrilling, funny, fast-paced contemporary reimagining of the Arthur Conan Doyle classic

▼ Jazyk: English	▼ Premiéra: 2010-07-25
Žánr: Drama	Délka: 90
Typ: Scripted	Ohodnocení: 9.2

Komentáře uživatelů



Komentář...

★★★★★

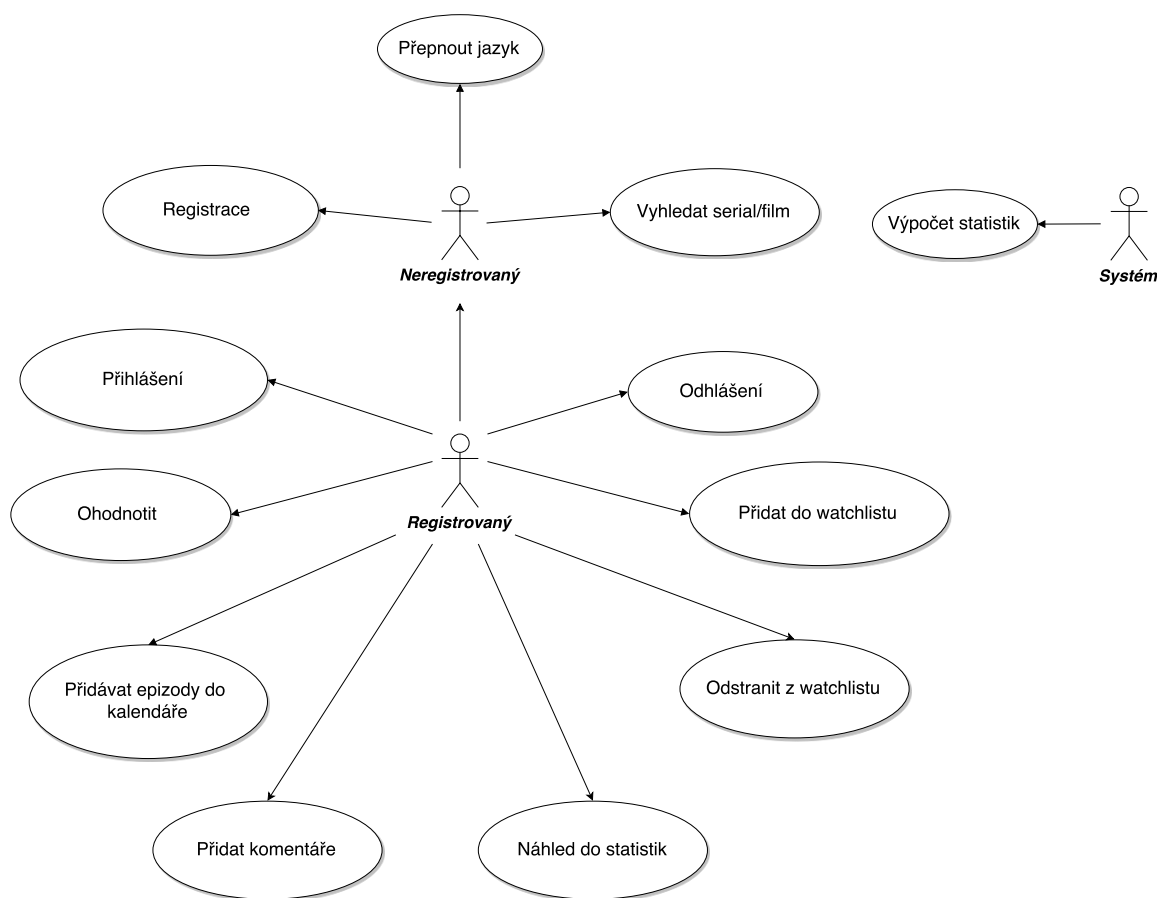
Přidat do watchlistu

Obrázek 3.2: Návrh detailů seriálu

3.3 Diagram případů užití

Na základě výše uvedených funkcí je vytvořen diagram případů užití (angl. Use-case diagram). Tento diagram nám znázorňuje interakci mezi uživatelem a systémem a nabízí základní možnosti pro jednotlivé role.

Diagram nepopisuje jakým způsobem budou akce prováděny, ale slouží k názornějšímu zobrazení různých možností poskytovaných systémem. Systém si lze v tomto případě představit jako černou skříňku (black box), jejíž obsah a detaily jednotlivých akcí není zvenčí viditelný. Z obrázku 3.3 lze vyčíst 3 základní role - registrovaný uživatel, neregistrovaný uživatel a systém. Také je tady vidět možnost dědění od neregistrovaného uživatele k registrovanému uživateli.

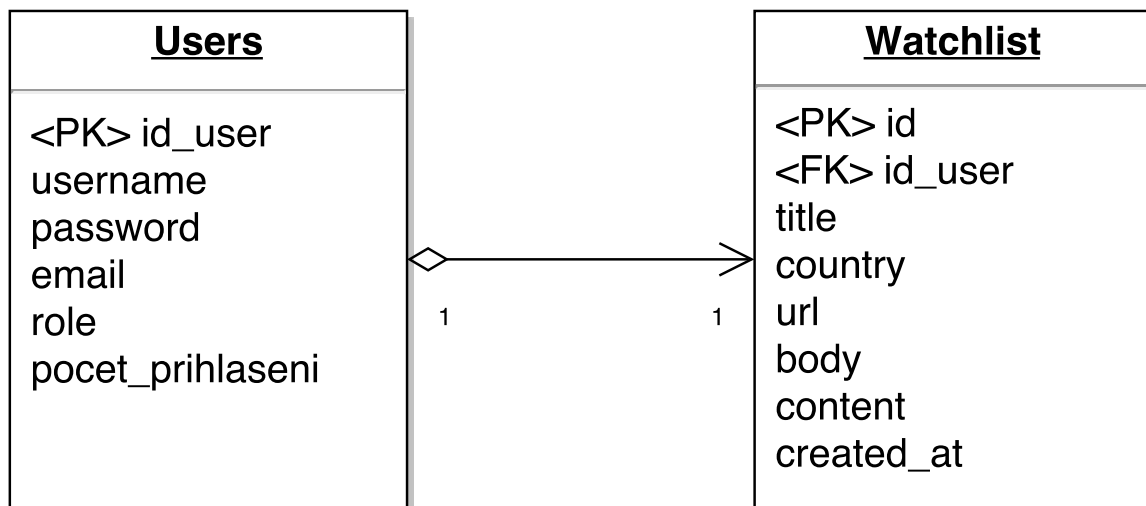


Obrázek 3.3: Diagram případů užití

3.4 Návrh databáze

Následujícím krokem pro návrh aplikace je vytvoření datového modelu. Na základě specifikace požadavků byl zvolen strukturovaný model návrhu, jehož zdrojem je ER diagram. ER diagram modeluje uložená data a vztahy mezi nimi.

Na obrázku 3.4 lze vidět, která data jsou uložena v databázi. Z požadavků systému plyne, že je nutné ukládat data o každém uživateli, který je registrovaný v systému.



Obrázek 3.4: ER diagram

Proto klíčovou tabulkou aplikace je tabulka *Users*. V této entitě jsou zásadní atributy *email*, *password* pro registraci a *username* a *password* pro další přihlašování do systému. Další tabulkou je *Watchlist*, která slouží pro ukládání seriálu do vlastního seznamu oblíbených filmů a seriálů. Tabulka obsahuje id uživatele (*id_user*) který si přidá seriál, název seriálu (*title*) a odkaz na stránku seriálu (*url*). Dalším atributem je *country*, která obsahuje informace o původním jazyku seriálu nebo filmu. Atribut *body* a *content* obsahují ohodnocení a komentáře uživatelů. Posledním atributem je *created_at*, který ukládá datum začátku sledování seriálu.

3.5 Architektura webové aplikace

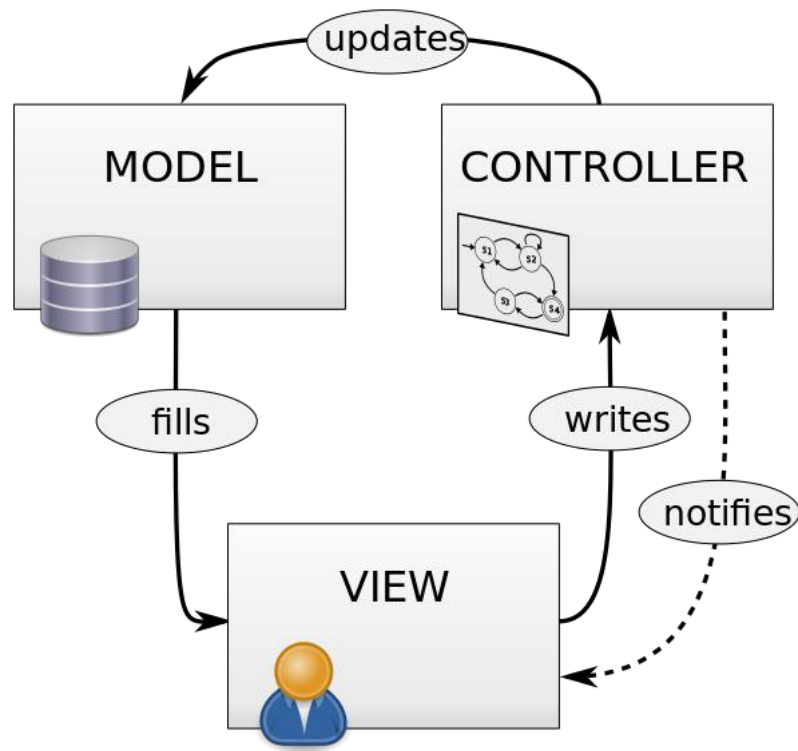
Pro kvalitní architekturu aplikace existují architektonické vzory. Architektonický vzor je opakovaně použitelné řešení běžně se vyskytujících problémů v softwarové architektuře, jsou podobné softwarovým návrhovým vzorům, ale mají širší rozsah. Většina webových aplikací je založena na architektuře *MVC*.

Model – View – Controller – je softwarová architektura, která rozděluje datový model aplikace, uživatelské rozhraní a řídicí logiku do tří nezávislých komponent tak, že modifikace některé z nich má jen minimální vliv na ostatní. Návrhový vzor *MVC* rozděluje program do tří hlavních částí [4]:

Model poskytuje data a metody které manipulují nad těmito daty. Modely především dochází k dotazování dat z/do databáze a kontrolu správnosti dotazovaných dat. Model není závislý na reprezentaci, neví jak vizualizovat data, nemá žádné body interakce s uživatelem - jednoduše poskytuje řízení a přístup k datům.

View je vrstva starající se o prezentaci dat (např. uživatelské prostředí) obsahuje přímý odkaz na model, aby mohl jeho data prezentovat vnějšimu světu.

Controller reaguje na události v programu, zajišťuje změny v pohledu nebo modelu a poskytuje spojení mezi uživatelem a systémem. Kontroluje a směřuje data od uživatele do systému a naopak. Propojení komponent lze vidět na obrázku 3.5.



Obrázek 3.5: Diagram MVC [5]

3.5.1 Použité technologie

Webové stránky zobrazují obsah a mají základní strukturu definovanou v jazyce HTML. Aplikace by měla mít příjemný vzhled, k tomu slouží jazyk CSS. Pomocí JavaScriptu můžeme měnit obsah webové stránky u uživatele, jazyk nabízí tvorbu dynamických prvků, a dalších kontejnerů, které umožňují ušetřit místo na stránce. Níže jsou popsány jednotlivé technologie:

HyperText Markup Language (HTML) název jazyka používaného pro tvorbu webových stránek, které jsou propojeny hypertextovými odkazy. Obsahem může být text, obrázky, odkazy nebo dokonce audio a video. Většina HTML elementů jsou zapsány pomocí otevření `<element>` a zavírání `</element>` tagu, které informují prohlížeč, jaký typ obsahu zobrazovat [18].

Cascading Style Sheets (CSS) je souborem metod, které umožňují měnit popis vzhledu elementů na stránkách napsaných v jazyku HTML. Jeho hlavním cílem je oddělení struktury a obsahu dokumentu od definice vzhledu. Jazyk popisuje způsob vykreslování, zarovnání, automatického formátování prvků [18].

JavaScript je objektově orientovaný skriptovací jazyk, používaný pro tvorbu dynamických webových stránek. Většinou se využívá na straně klienta, tím dokáže reagovat na interakci uživatele bez opakovaného načtení stránky. Program se spouští souběžně s webovou aplikací, na rozdíl od ostatních interpretovaných programovacích jazyků (např. PHP), který se nejprve provede na serveru a potom odešle prohlížeči pouze výsledek [20].

Hypertext Preprocessor (PHP) [17] slouží pro implementaci serverové části aplikace. Oblibu si tento jazyk získal díky podporování hodně knihoven pro různé účely – zpracování textu, grafiky, práci se soubory, přístup k databázovým systémům (MySQL, Oracle). Tvorba webové aplikace při použití samotného jazyka PHP je příliš časově náročná. Z toho důvodu začaly vznikat různé frameworky, které mají za úkol usnadnit vytváření základních funkcí aplikace např.: registrace nebo přihlašování uživatelů, práce s databází atd. . .

Ve světě je velice oblíbeným frameworkem pro vývoj webových aplikací Laravel. V Česku se vývojáři přiklání k domácímu frameworku Nette. Zpočátku psaní aplikace pomocí frameworku je pomalé a obtížné, je potřeba se seznámit s frameworkem, naučit se implementovat základní funkce které nám nabízí dokumentace nebo tutoriál. Po úspěšném pochopení a naučení se všech konstrukcí a principů, které se v daném frameworku vyskytují, je vývoj webových aplikací velice rychlý a efektivní.

My Structured Query Language (MySQL) je jeden s nejpopulárnějších systémů řízení báze dat. MySQL je volně dostupný relační databázový systém, který vychází z deklarativního programovacího jazyka SQL (Structured Query Language). Jedná se o nejrozšířenější systém, který klade důraz na maximální efektivitu a rychlost. Často je nasazován v kombinaci s operačním systémem Linux, Apache, MySQL, za přítomnosti interpretů jazyka PHP [12]. Pro jednodušší správu databáze je možné použít phpMyAdmin.

3.5.2 Framework

Pro realizaci webové aplikace byl vybrán framework Nette. Jedná se o framework podporující *DRY*, *KISS* a používající MVC strukturu a objektové programování s velkým důrazem na bezpečnost aplikace. Obsahuje mnoho ladících nástrojů, které pomáhají programátorovi tvořit bezchybný kód. Pro práci s databází využívá Nette Database, *NotORM* nebo *Dibi* knihovny, které podporují téměř všechny nejpoužívanější databázové formáty. Vlastnosti frameworku [16]:

- kvalitní ladící nástroje
- podpora v českém jazyce
- rychlá práce s databází
- srozumitelný a přehledný kód
- podpora HTML 5, SEO, AJAX
- dobré zabezpečení

Dokumentace je jasně a přehledně napsána a pro českého programátora je výhodou, že je napsaná v českém jazyce. V dokumentaci je obsaženo velké množství ukázek a příkladů, které značně zrychlí pochopení frameworku [14]. K dispozici je stránka *Planette* [13] která poskytuje velké množství různých návodu a ukázkových kódů.

3.6 Vybrané API

V této sekci jsou popsány použité API a jejich možnosti. Zpočátku se podíváme na API *AddEvent* pro přidání jednotlivých epizod seriálu do kalendáře. Dále jsou popsány možnosti API od *IMDb* a *TVMaze*.

3.6.1 AddEvent

AddEvent je společnost založena v roce 2012 a dnes je nejpopulárnější ve své oblasti. AddEvent poskytuje API ve formě “widgetu” který se skládá z tlačítka “Přidat do kalendáře” a seznamu nabízejících kalendářů: Apple Calendar, Google, Outlook, Yahoo. API poskytuje jednorázové přidání akce do kalendáře které je bezplatné. Webová aplikace nabízí premium účty kde je možnost hromadného přidání akcí do kalendáře, v ceně 58\$ ročně. Premium účty taky zahrnují rozšířené možnosti např.: změna css stylu, pokročilé nastavení javascriptu, SSL podporu [15].

Základní možnosti zobrazených informací v kalendářů je možné vidět v níže uvedeném kódu:

```
1 <div title="Add to Calendar" class="addeventatc">
2 Add to Calendar
3 <span class="start">05/05/2017 09:00 AM</span>
4 <span class="end">05/05/2017 11:00 AM</span>
5 <span class="timezone">Europe/Prague</span>
6 <span class="title">Summary of the event</span>
7 <span class="description">Description of the event<br>Example of a new line</span>
8 <span class="location">Location of the event</span>
9 <span class="organizer">Organizer</span>
10 <span class="organizer_email">Organizer e-mail</span>
11 <span class="facebook_event">https://www.facebook.com/events/150586325277</span>
12 <span class="all_day_event">false</span>
13 <span class="date_format">MM/DD/YYYY</span>
14 <span class="client">apHXbftvZznVlJNSJmVh25714</span>
15 </div>
```

3.6.2 IMDb

The Open Movie Database je bezplatná webová služba k získání informací o filmech. Umožňuje vyhledat film podle ID nebo názvu s určitými parametry [19]:

1. Rok vydání.
2. Návratovy datový typ.
3. Stručný nebo rozšířený popis filmu.

Nalezený film obsahuje základní informace o filmu a jejich detaily: název, rok, premiéra, délka, žánr, autoři, herci, popis filmu, jazyk, země, plakát a ohodnocení uživatelů. Data-báze *IMDb* byla zvolena zejména kvůli svým objektivním hodnocením uživatelů a všech potřebných informací o filmu.

3.6.3 TVMaze

Pro vývojáře databáze nabízí bezplatné REST API které je snadno ovladatelné. REST je architektura rozhraní, která umožňuje přistupovat k datům pomocí metod *HTTP* [8]. Výstupem

jsou data ve formátu vracející JSON které odpovídají HATEOAS a HAL principům, které se odlišují od ostatních síťových aplikačních architektur. Princip spočívá v komunikaci klientské aplikace s aplikačním serverem skrze dynamický poskytovaný odkaz. TVMaze API bylo zvoleno především z důvodů svoje popularity ostatními seriálovými databázemi. Dalším důvodem je předchozí používání dané databáze která zdarma poskytuje téměř všechny informace na webu [9]:

Show search	hledá v databázi všechny pořady podle názvu. Je využíván Fuzzy algoritmus s hodnotou nepřesnosti 2, což znamená, že pořady budou nalezeny i když dotaz obsahuje malé překlepy.
People search	vyhledává všechny lidi v databázi, využívá stejný algoritmus, popsaný výše.
Shows	načítá všechny základní informace o daném pořadu.
Show seasons	kompletní seznam sezón pro daný pořad.
Show episode	poskytuje kompletní seznam epizod pro daný pořad. Epizody jsou vráceny v jejich vysílaném pořadí a zahrnuje rozšířenou informaci o epizodách.
Show crew	vrátí seznam hlavní posádky pro pořad.
Show cast	vrátí seznam hlavních herců. Každá položka herců je kombinací osoby a postavy.
Show index	ukáže seznam všech nejlepších pořadů v databázi.
Show lookup	vyhledá přesný pořad při znalosti, tvrage , thetvdb nebo IMDb ID. V případě, totožnosti, vrátí HTTP 302 přesměrování na adresu URL. V opačném případě je odeslána HTTP 404.
Schedule	je kompletní seznam epizod v dané zemi k danému datu. Epizody jsou vráceny v pořadí, v jakém jsou vysílané úplné informace o epizodě a odpovídajícímu pořadu.
Full schedule	je seznam všech dalších epizod bez ohledu na jejich zemi. Výsledky jsou ukládány do mezipaměti po dobu 24 hodin.
Episode by number	načte konkrétní epizodu s uvedeným číslem sezony a epizody. To buď vrátí úplnou informaci pro jednu epizodu, nebo HTTP 404.
Episodes by date	načte všechny epizody vysílaných v určitém datu. Vrátí podrobný popis epizody, nebo HTTP 404.
Show AKA's	vrátí seznam názvu pořadu v jiných zemích.

Kapitola 4

Implementace

V této kapitole je popsána samotná realizace aplikace. Každá z podkapitol je zaměřena na způsob řešení podstatných částí aplikace na základě specifikace a předchozího návrhu. Nejprve se soustředíme na serverovou část a potom na databázový systém. Vývoj probíhal na vlastním počítači s operačním systémem *MacOS* ve vyvojevě prostředí *PhpStorm 2016.1* od společnosti IntelliJ. Serverová část aplikace je naprogramovaná ve skriptovacím jazyku *PHP 5.6* s využitím frameworku *Nette 2.4*. Grafický vzhled systému byl vytvořen pomocí frameworku *Mobirise 3.1* s *HTML5* a *CSS3*.

4.1 Serverová část

Tato část se věnuje získání dat pomocí použitých API. Nejprve bylo nutné vytvořit funkci pro získání dat z databáze a následně je vyparsovat. Jednotlivé části jsou popsány níže.

4.1.1 Vyhledání

Na základě návrhu a požadavků systému, popsaných v sekci 3.1 nejprve bylo nutné vytvořit funkci pro získání dat z databáze. Vyhledáno informací o konkrétním seriálu nebo filmu zajišťuje vyhledávací formulář, který je rozdělen na 2 části, v závislosti na položce které uživatel vyhledává. První část má na starosti vyhledání seriálu podle zadaného názvu. Druhá část se zas stará o vyhledání filmu. Oba druhý vyhledávaných položek jsou zpracovávány jedním formulářem jehož implementační detaily lze vidět níže.

```
1 <?php
2 public function searchFormSucceeded(UI\Form $form, $values){
3     $data = $form->getValues();
4     if ($data->typ === 'serial'){
5         $tvmaze = new TVMaze();
6         $this->serial = $tvmaze->search($data['name']);
7         $_SESSION['serial'] = $this->serial;
8         $this->redirect('Homepage:serial');
9     }
10    else if($data->typ === 'film'){
11        $omdb = new OMDB();
12        $this->imdb = $omdb->search($data['name']);
13        $_SESSION['imdb'] = $this->imdb;
14        $this->redirect('Homepage:films');
15    }
16 }
17 ?>
```

Zdrojový kod 4.1: Formulář pro vyhledání filmů a seriálů

Zádaná data z formuláře zpracovává metoda *searchFormSucceeded* implementována v presenteru *Homepage*. Vytvoříme instanci třídy *TVMaze* pro seriály a *OMDb* pro filmy. Nejprve se použije metoda *search()* kterou je možné vidět na výše uvedeném zdrojovém kódu na řádce 6 a 12. Dále se data zpracují a uloží do *Session*. Následně dojde k přesměrování uživatele na stránku, kde už budou zobrazeny konkrétní informace o vyhledávaném filmu nebo seriálu. Obsahem stránky, na kterou bude uživatel následně přesměrován, jsou data ve formátu *JSON*. *JSON* je následně dekodován pomocí funkce *json_decode*. Takto dekodované data jsou uložena do vícerozměrného pole pro jejich lepší manipulaci. Data uložená v poli jsou nakonec zobrazeny na stránce na kterou je uživatel přesměrován.

```
1 function search($jmeno) {  
2     $url = 'http://api.tvmaze.com/search/shows?q=' . urlencode($jmeno);  
3     return json_decode(file_get_contents($url), true);  
4 }
```

Zdrojový kód 4.2: Funkce pro získání dat pomocí API

4.1.2 Parsování

Po nalezení filmu nebo seriálu s podobným názvem, je potřeba z pole vyparsovat informaci. Každý nalezený seriál je pole, které se skládá z dalších polí. Pole obsahují plakáty, různé informace o seriálu/filmu a *url* na hledaný seriál nebo film.

Nejdřív bylo nutné zpracovat obsah pole a dostat z ní potřebné informace ve formě textových řetězců tak, abychom je mohli správně zobrazit na stránce. Následně byly v cyklu rozparsovány všechny seriály které tvořily obsah pole. U jednotlivých seriálů/filmů byl vykreslen pouze plakát, název a odkaz na sebe. Na obrázku 4.1 a 4.2 je vidět rozdíl mezi získanou odpovědí, její vyparsováním a vykreslením.

Na stránku jednotlivého seriálu se dostaneme pomocí odkazu, který se nachází v poli `['_links']['self']`. Funkce *actionEpisode()* pomocí metody *file_get_contents()* získá obsah stránky a zkonvertuje ho do pole. Ke konverzi obsahu do pole je opět použita funkce *json_decode()*. Stejný princip přesměrování se využívá i při dalších podstránkách.

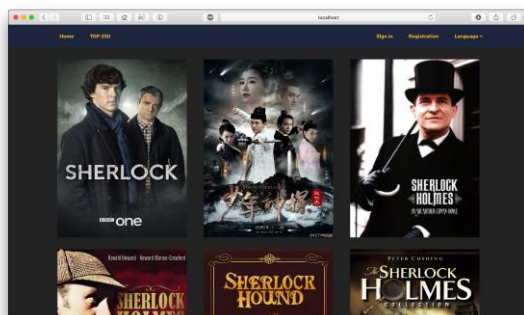
Finální stránka, která je uživateli nakonec zobrazena obsahuje plakát, popis a různé další informace o seriálu. Ty jsou uvedeny v tabulce. Detail filmu dále obsahuje komentáře jiných uživatelů a informace o počtu lidí, kteří tento seriál sledují. Pro zaregistrované uživatele je zobrazen formulář který obsahuje vstupní pole typu `texttt` `textarea`. Vstupní pole toho typu není omezené počtem znaků. Uživatel tedy může přidat libovolné množství znaků a vyjádřit tak svůj názor na daný film či seriál. Součástí formuláře je i *selectbox*, který obsahuje hodnoty na stupnici od 1 do 10, kde 1 je nejhorší a 10 nejlepší možná udělena známka. Uživatel musí vybrat jednu z těchto hodnot a udělit tak své hodnocení filmu nebo seriálu. Po stisknutí tlačítka “Přidat do watchlistu” uživatel uloží vlastní komentář s ohodnocením a přidá seriál do vlastního seznamu sledovaných seriálů. Způsob ukládání informací a přidání do seznamu je popsán v databázové části.

Vrchol stránky obsahuje odkazy na rozšířené informace o seriálu. V případě že si uživatel chce přehlédnout epizody seriálu, je na stránce k dispozici odkaz “sezóny”. Na dané podstránce má uživatel k dispozici všechny dostupné sezóny daného seriálu. Obsahem stránky je popis, plakát, země seriálu, počet epizod, kanál a taky datum úvodní a poslední části. Jednotlivé sezóny jsou získány odkazem který je zobrazen na obrázku 4.4.

Stejný princip odkazu má podstránka “epizody”. Obsahem podstránky je seznam všech epizod daného seriálu. Podstránka obsahuje informaci o čísle epizody v rámci sezóny, název



Obrázek 4.1: Pole nalezených seriálů



Obrázek 4.2: Vykreslení seriálů

a plakát. Po otevření konkrétní epizody uživatel navíc obdrží popis a datum zveřejnění epizody. Posledním tlačítkem je odkaz na upoutávku seriálu dostupnou na stránce YouTube¹.

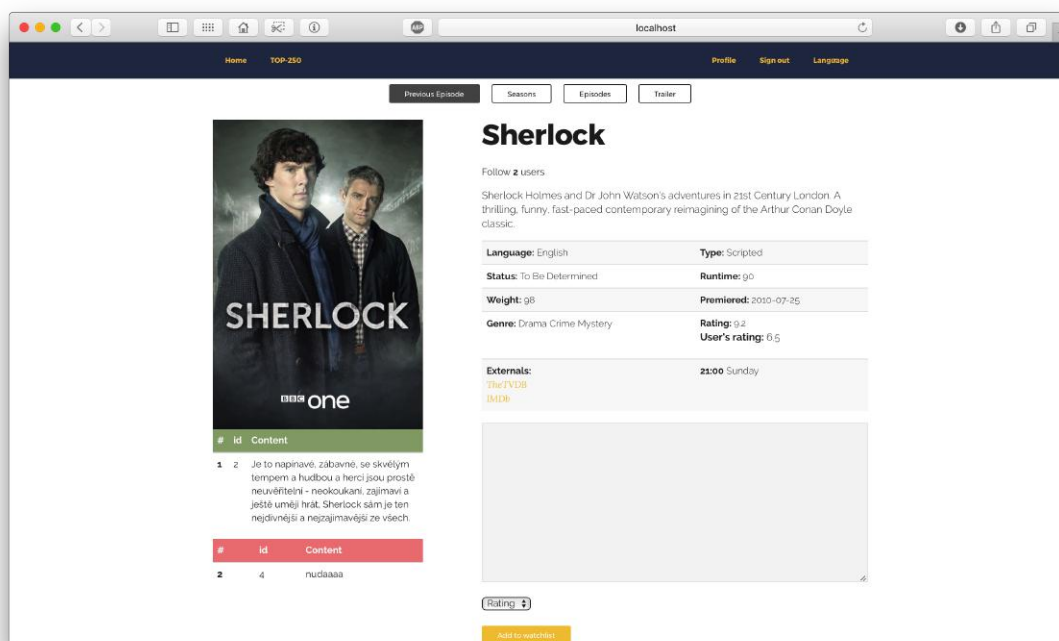
Aby uživatel nemusel hledat které epizody jsou právě vysílané, bylo nutně naimplementovat zobrazení aktuálních epizod. Pro jejich zobrazení je potřeba z pole vybrat všechny data. Jedná se hlavně o informace jako název a číslo epizody, čas jejího zveřejnění, plakát a číslo sezóny v níž daný seriál vyšel.

Následně pomocí funkce `renderAllEpisodes()` zjistíme aktuální datum. K tomu slouží funkce `date()` pomocí které nastavíme potřebný formát `Y-m-d` ve kterém jsou všechna data epizod *TVMaze API*. Potom provedeme porovnání aktuálního data s daty všech epizod. Pokud datum epizod je větší než aktuální datum, vykreslíme vysílané epizody na vrcholu stránky. Taky bylo nutné pro každou aktuální epizodu změnit formát data na `d/m/Y` kvůli omezení podpory různých formátů aplikací *AddEvent*.

4.1.3 Přidání do kalendáře

Přidání do kalendáře je dostupné na podstránce *homepage/next* pokud chceme přidat další epizodu a *homepage/all-episodes* pokud chceme přidat všechny další vysílané epizody. Pro přidání API do vlastní aplikace byla potřeba do hlavičky *html* souboru připojit knihovnu. Ve funkci `renderAllEpisodes()` vytvoříme dvě pole `$arr` a `$arr1`. Do pole `$arr1` vložíme všechny potřebné informace které chceme vidět v kalendáři: název seriálu a jednotlivé epizody, datum a čas zveřejnění epizody, číslo sezóny a epizody v rámci sezóny a popis epizody.

¹youtube.com



Obrázek 4.3: Zobrazení seriálu

<http://api.tvmaze.com/shows/335/seasons/>

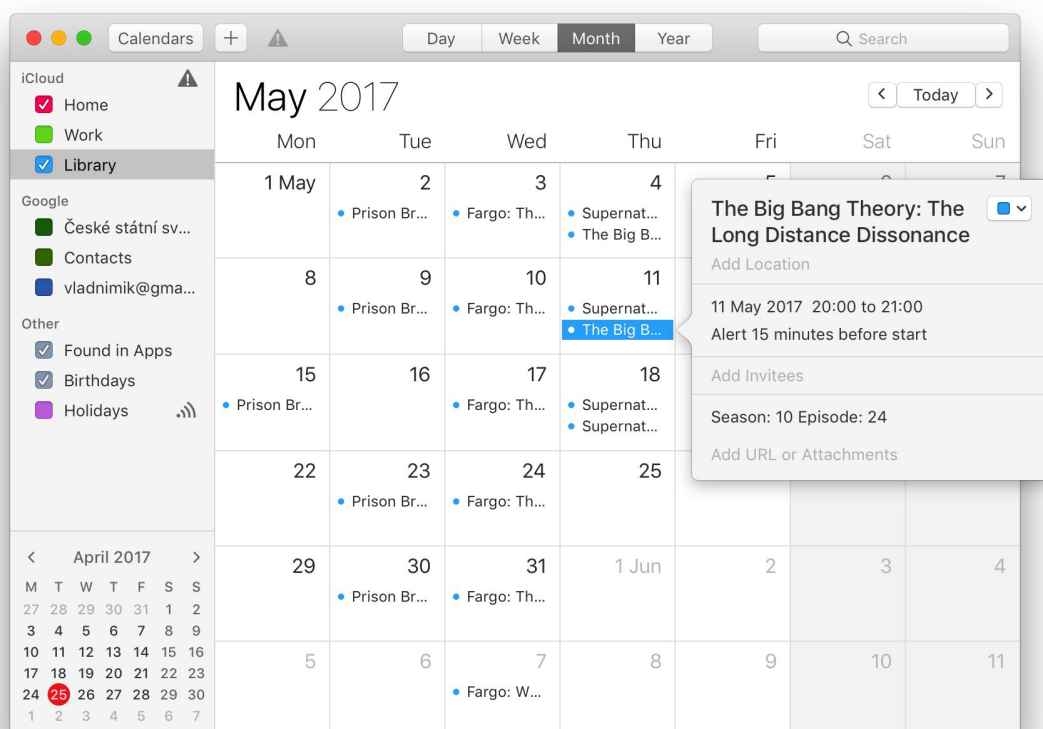
— všechny seriály

— id seriálu

— všechny sezony daného seriálu

Obrázek 4.4: Odkaz na všechny sezony seriálu

Následně v cyklu, do pole `$arr` pomoci funkce `array_push()`, která v jazyku *php* umožňuje používat pole jako zásobník, vložíme hodnoty pole `$arr1`. Následně si uživatel může vybrat který kalendář chce použít. Má na výběr několik možností: Apple Calendar, Google Calendar, Outlook a Yahoo. Při výběru online kalendářů, si uživatel může upravit nebo přidat vlastní poznámky a uložit je. Při výběru *Apple Calendar* počítač stáhne kalendář ve formátu **.ics* a otevře ho v aplikaci *Calendar*. Před 15 minutami zveřejnění epizody, uživatel obdrží upozornění na zařízení které je synchronizováno s kalendářem. Na obrázku 4.5 je možné vidět zobrazení kalendáře přidanych epizod a jejich popis.



Obrázek 4.5: Kalendář přidaných epizod seriálů

4.2 Databazová část

Po realizaci serverové části přichází na radu další část a tou je realizace databázového systému. Tato sekce se věnuje samotnému připojení do databáze, přihlašování uživatelů a hodnocení filmů/seriálů, přidávání sledovaných filmů do seznamu a vytvoření různých statistik.

4.2.1 Připojení k databázi MySQL

Ze začátku bylo nutné vytvořit databázi pro uložení dat. Databáze obsahuje pouze dvě tabulky protože aplikace využívá externí databázi ke které přistupujeme pomocí API. Tím pádem není potřeba ukládat žádné informace o filmech a o seriálech do vlastní databáze. Databáze byla vytvořena podle návrhu datového modelu uvedeného v kapitole 3.3. K vytvoření databáze byl použit nástroj *phpMyAdmin* umožňující jednoduchou správu obsahu databáze a poskytující jednoduché uživatelské rozhraní. Aby se dalo s databází pracovat, bylo nutné se k ní připojit. Připojení pomocí frameworku *Nette* je velmi jednoduché, stačí správně nastavit konfigurační soubor. Příklad konfiguračního souboru je uveden níže:

```
1 database:
2   dsn: 'mysql:host=localhost;dbname=filmoteka'
3   user: 'root'
4   password: 'root'
```

```

5 | options:
6 |     lazy: yes

```

Zdrojový kod 4.3: Konfigurace databáze

Na výše uvedeném příkladu je možné vidět nastavení adresy databázového serveru a taky jméno databáze. Dále je uvedeno přihlašovací login a heslo. Dalším parametrem je parametr *lazy*, který udává, že spojení s databází se provede, až pokud to bude zapotřebí.

4.2.2 Přihlašování uživatelů

Pro umožnění využití pokročilejších funkcí aplikaci je nutné se přihlásit. Uživatel se přihlásí pomocí přihlašovacího formuláře na adrese `sign/in`. K vykreslení formuláře slouží komponenta *SignInForm*. Pro přihlášení je uživateli zobrazen formulář, který musí být pro odeslání vyplněn.

Obrázek 4.6: Formulář pro přihlášení

Vyplněná data z formuláře zpracovává metoda `signInFormSucceeded()`. Ta se nachází v presenteru *Sign*. Následně data se předají funkci `login`. Toto je vidět na níže uvedeném zdrojovém kódu na řádce 7. Metoda následně předá data k autentizaci metodě *authenticate*, která vyhledá žádaný login v tabulce *users*. Pokud login existuje, porovná zadané heslo s heslem uloženým v databázi. Hesla jsou zašifrovaná pomocí šifrovací funkce *hash* ve formátu *sha1*. Následně byly porovnány hašované zadané heslo s hašovacím heslem v tabulce.

```

1 | public function signInFormSucceeded(Form $form){
2 |     $values = $form->getValues();
3 |     $user = $this->getUser();
4 |     $authenticator = new \auth\MyAuthenticator($this->database);
5 |     $user->setAuthenticator($authenticator);
6 |     try {
7 |         $user->login($values['username'], $values['password']);
8 |         $user->setExpiration('20 minutes', TRUE);
9 |         $this->redirect('Homepage:default');
10 |     }
11 |     catch (Nette\Security\AuthenticationException $e) {
12 |         $form['password']->addError('Wrong password or username');
13 |     }
14 | }

```

Zdrojový kod 4.4: Formulář pro přihlášení uživatelů

Registrace uživatelů byla vyřešena podobným způsobem. U registračního formuláře uživatel navíc musí zadat *e-mail*, pomocí kterého se zbavíme duplicitních uživatelů.

4.2.3 Přidávání do watchlistu

Jak bylo uvedeno výše, formulář pro přidávání seriálu do vlastního seznamu se nachází na stránce jednotlivého seriálu. Pro přidání seriálu do vlastního watchlistu je potřeba ho ohodnotit a podle libosti okomentovat. Pokud se uživatel rozhodne přidat komentář, aplikace ho automaticky přidělí do ostatních komentářů daného seriálu a taky vykreslí na řádku seriálu ve vlastním seznamu. Komentáře jsou rozděleny na 2 části, podle přidělených bodů. Komentář uživatele, který obsahuje hodnocení pět a výše bodů je graficky zvýrazněn zelenou barvou. Daně hodnocení je tak vnímáno jako pozitivní. V opačném případě je komentář považován za negativní a jde tedy o uživatelskou kritiku. Příklad je vidět na obrázku 4.3

Ve funkci `renderEpisode()` zjistíme *url* daného seriálu a uložíme ho do *session*. Po stisknutí tlačítka “Přidat do watchlistu” se zavolá metoda `commentFormSucceeded()`.

```
1 public function commentFormSucceeded($form, $values)
2 {
3     $user = $this->getUser()->getIdentity()->getId();
4     $sess = $_SESSION['link'];
5     $exists = $this->database->table('watchlist')->where('id_user',$user)->where('url', $sess)->
        fetchAll();
6     if ($exists == NULL){
7         $this->template->watchlist = $this->database->table('watchlist')->insert(array(
8             'title' => $_SESSION['episode']['name'],
9             'country' => $_SESSION['language'],
10            'url' => $_SESSION['link'],
11            'id_user' => $this->getUser()->getIdentity()->getId(),
12            'body' => $values->body,
13            'content' => $values->content,
14        ));
15        $this->redirect("Profile:default", array("ID" => $user));
16    }
17 }
```

Zdrojový kod 4.5: Formulář pro přidávání seriálů do watchlistu

Nejprve zjistíme *id* uživatele který si přidává seriál do vlastního watchlistu. Následně si ze *session* vytáhneme *url* stránky. Na řádku 5 pomoci dotazů ověříme, zda v databázi existuje seriál se stejným *url* od stejného uživatele. Pokud v databázi žádný není, do tabulky *watchlist* přidáme název daného seriálu, země, id uživatele, přidělené body a komentář. Po odeslání formuláře systém přesměruje uživatele na stránku */profile* ve které provede konkatenaci *id* uživatele, což je vidět na řádku 15. Ve vlastním profilu má uživatel k dispozici tabulku přidáných filmu a seriálu. Obsahem tabulky je název filmu nebo seriálu, ohodnocení, komentář a datum začátku sledování. Datum začátku sledování seriálu je určen při přidání seriálu do vlastního watchlistu. Čas vzniku vyjadřuje atribut *created_at* typu *timestamp* ve formátu (F j , Y), kde F označuje den, j měsíc a Y rok. Pokud uživatel už nechce sledovat seriál nebo film, má možnost smazat ho z watchlistu. Hodnocení a komentář se odstraní pod jednotlivým seriálem/filmem automaticky.

4.2.4 Statistiky

Dalším krokem implementace jsou různé statistiky aplikace. Odkaz na statistiky se nachází na stránce profilu registrovaných uživatelů. Samotné statistiky jsou dostupné na adrese */stats*. Podle statistik uživatele můžou vidět zájmy ohledně seriálů a filmů ostatních uživatelů. Uživatelé mají k dispozici informací nejpopulárnějších seriálech podle hodnocení uživatelů daně aplikace. Statistiky taky obsahují informace o počtu uživatelů které si přidali jednotlivé filmy a seriály do vlastního watchlistu. Dále obsahují statistiky informace o země původu, kde byl film nebo seriál natočen.

Nejprve v presenteru *Sign* ve funkci `signInFormSucceeded()` byla naimplementovaná statistika o počtu přihlášení jednotlivých uživatelů. Z databáze bylo potřebné dostat *login* a *počet_přihlášení*. Do `sessionSection` bylo uloženo číslo přihlášení jednotlivého uživatele, které následně bylo předáno proměnné `$počet`. Proměnná `$pocet` v tomto případě slouží jako počítadlo. Pokud se *id* uživatele shodovalo s *id* uživatele v tabulce *users*, dojde k inkrementaci atributu *počet_přihlášení* u daného uživatele o jedničku. Následně byly všechny informace o počtu přihlášení vypsané sestupně na stránce statistik.

```

1 $this->template->rating = $this->database->query('
2 SELECT title,
3 AVG(body) FROM watchlist
4 GROUP BY title
5 ORDER BY body DESC ')->fetchAll();
6
7 $this->template->quantity = $this->database->query('
8 SELECT id_user,
9 COUNT(id_user) mycount
10 FROM watchlist
11 GROUP BY id_user
12 ORDER BY mycount DESC')->fetchAll();
13
14 $this->template->title = $this->database->query('
15 SELECT title,
16 COUNT(title) mycount
17 FROM watchlist
18 GROUP BY title
19 ORDER BY mycount DESC')->fetchAll();
20
21 $this->template->country = $this->database->query('
22 SELECT country,
23 (COUNT(country)* 100.0 / (
24     Select COUNT(*)
25     From watchlist)) as mycount
26 FROM watchlist
27 GROUP BY country
28 ORDER BY mycount DESC');

```

Zdrojový kod 4.6: SQL dotazy pro zobrazení statistik

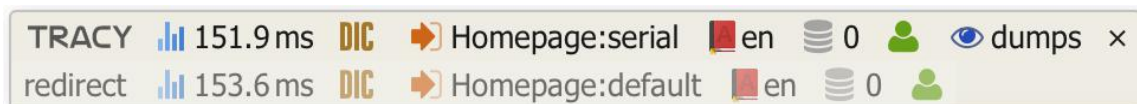
Další statistikou byl žebříček nejlepších filmu a seriálu registrovaných uživatelů. Pomocí *SQL* dotazů byly vybrány názvy všech hodnotících filmů a seriálů a spočítán průměr bodů. Následujícím dotazem na řádku 7 bylo zjištěno kolik filmu a seriálu ve watchlistu mají jednotlivé uživatele. Na řádku 14 je dotaz ve kterém vypisujeme všechny filmy a seriály a počet uživatelů které je sledují. Tím pádem zjistíme nejpoblárnější filmy a seriály naše aplikace. Na stránce jednotlivého seriálu vypíšeme informaci o počtu uživatelů který daný seriál sledují. Dalším dotazem byla potřeba zjistit nejpoblárnější jazyk sledovaných seriálu. Dotaz je vidět na řádku 21. Parametr `country` v tomto dotazu znamená “země původu” seriálu, což se shoduje s původním jazykem ve kterém byl seriál natočen. Výsledkem budou různé jazyky seriálu a jejich procentuální rozložení.

Kapitola 5

Testování

Testování aplikace probíhalo v průběhu samotného vývoje. Po každé dokončené funkci se provedl její test. Po dokončení aplikace bylo provedeno testování všech funkcí jako celku. Právě při testování bylo naraženo na chybu při implementaci hodnocení filmu a seriálu. Uživatel mohl přidat film a seriál do watchlistu bez jeho hodnocení, které je povinné. Na základě této chyby docházelo k poškození statistik. Další chybou bylo to, že při vrácení, ze stránky samotného seriálu nebo filmu se zobrazovaly pozdější požadavky hledaných seriálu. Tato chyba nastala z důvodů totožného *url*. Následně byly všechny chyby odstraněny.

Nette framework nabízí vlastní testovací nástroj, který byl využit od začátku implementování aplikace. Tím nástrojem je *Tracy*. První, čeho si můžete na stránce všimnout je *Debugger Bar*. To je plovoucí panel, který se zobrazí v pravém dolním rohu stránky. *Debugger Bar* obsahuje informaci o přihlášených uživatelích, o času načítání stránky a o akci která momentálně probíhá v presenteru.



Obrázek 5.1: Debugger Bar

Tracy nahrazuje chybové hlášení jazyka PHP, který oznamuje chyby přímo do zdrojového kódu stránky. Při použití debuggeru se nám zobrazí typ chyby, řádek na kterém vznikla a její podrobnější specifikace.

5.1 Návrh testovacího formuláře

Cílem bylo vytvořit srozumitelné otázky a jasné úkoly pro účastníky testování. Nejprve bylo nutné seznámit uživatele s vytvořenou aplikací a navrhnout úkoly ve správném pořadí. Po splnění úkolů, uživatelé měli odpovědět na otázky ohledně různých částí aplikací a splněných úkolů. Jednotlivé otázky a úkoly jsou uvedeny níže:

- Pro vyhledání filmu a seriálu, jakou databázi používáte?
- Stalo se vám někdy, že jste zapomněli na novou epizodu?
- Úkol: Zaregistrujte se do systému a následně se taky přihlaste.
- Úkol: Vyhledejte oblíbení seriál nebo film.

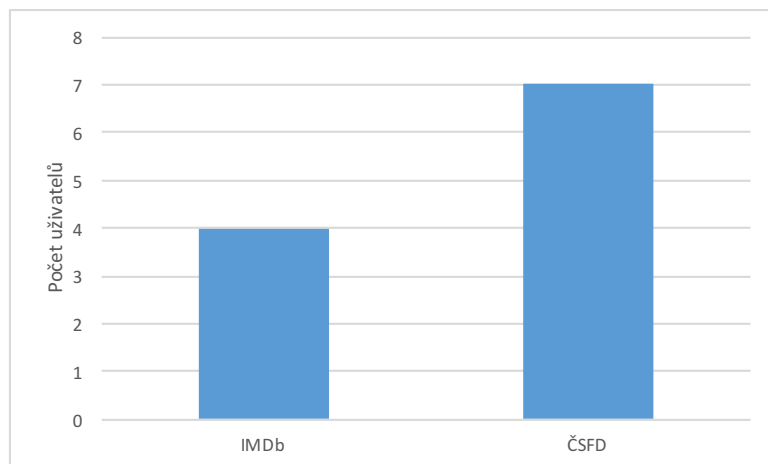
- Úkol: Ohodnoťte seriál a přidejte do watchlistu.
- Úkol: U vysílaného seriálu přidejte nové epizody do kalendáře.
- Které informace na stránce seriálu vám chybějí?
- Jak hodnotíte funkci přidání epizody do kalendáře?
- Jak se Vám líbil design aplikace?
- K čemu byste tuto aplikaci použil?

5.2 Finální testování

K tomuto testování byl vytvořen formulář pomoci služby *Google Forms*, která poskytuje možnost vytváření dotazníků a jejich vyplňování uživateli. Testování se zúčastnilo 11 osob různých věkových kategorií. Níže jsou uvedeny odpovědi účastníků testování na otázky uvedené ve formuláři.

Jakou databázi používáte pro vyhledání filmu a seriálu?

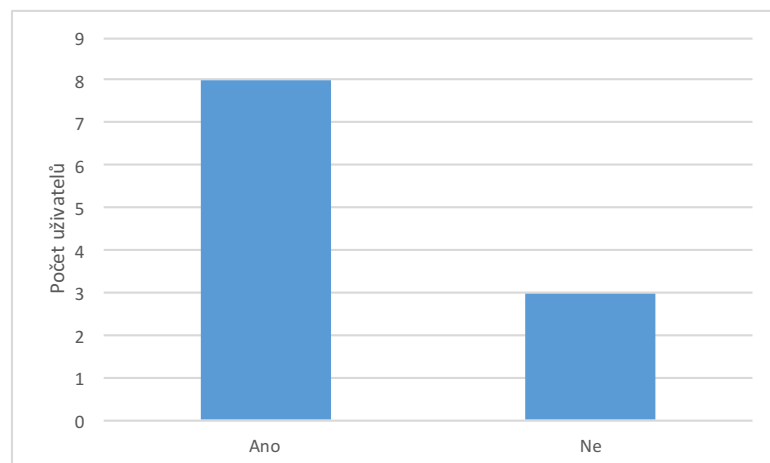
Většina uživatelů uvádí, že pro vyhledávání filmu a seriálu využívají česko-slovenskou databázi *ČSFD*. Druhou nejnavštěvovanější mezi účastníky testování je databáze *IMDb*.



Obrázek 5.2: Graf používaných databází

Stálo se vám někdy že jste zapomněl na novou epizodu?

Tato otázka je především určena pro uživatele, které sledují nové epizody vysílajícího seriálu. Cílem otázky bylo zjistit jaká část účastníků zapomíná sledovat nové epizody kvůli absenci upozornění. Získanou odpovědí je to, že 8 účastníků zapomínají na nové epizody seriálu a 3 účastníků sledují nové epizody pravidelně.



Obrázek 5.3: Graf sledování epizod

Úkol: Zaregistrujte se do systému a následně se taky přihlaste.

U tohoto úkolů museli uživatelé vyplnit registrační formulář a následně je doplnit do přihlašovacího formuláře. S registrací a přihlašováním do systému neměl nikdo problém.

Úkol: Vyhledejte oblíbení seriál nebo film.

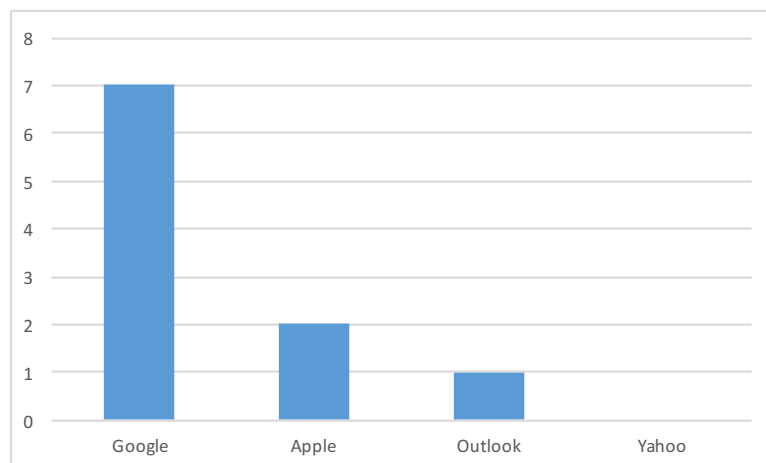
U tohoto úkolů taky uživatelé neměli žádné problémy. Navíc jeden uživatel poznamenal, že byl překvapený tomu, že v aplikaci se podařilo najít zahraniční seriál který na rozdíl od oficiální stránky obsahoval datum premiéry nové sezóny.

Úkol: Ohodnoťte seriál a přidejte do watchlistu.

Ani tento úkol nebyl pro žádného uživatele problémovým a každý ho dokázal zvládnout.

Úkol: U vysílaného seriálu přidejte nové epizody do kalendáře.

Tento úkol pro účastníky testování nedělal žádný problém. Většina lidí přidali nové epizody do kalendáře *Google*. Druhým nejpoužívanějším kalendářem je *Apple Calendar* a za ním následuje *Outlook*. Kalendář od *Yahoo* nepoužil ani jeden z účastníků testování.



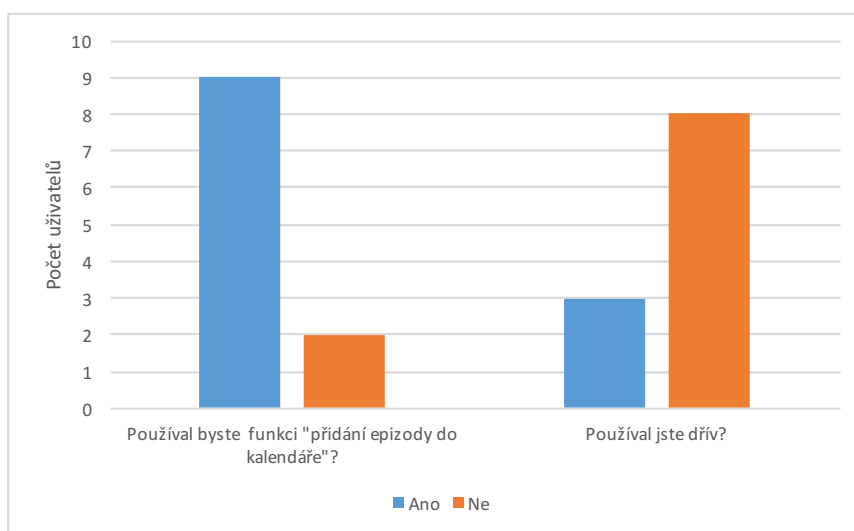
Obrázek 5.4: Graf nejpopulárnějších kalendářů

Které informace na stránce seriálu vám chybějí?

Některý z účastníků zaznamenal chybějící pokročilé informace o hercích. Problémem je absence rozšířené informace o hercích v databázi *TVMaze* a absence jakékoli informací v *OMDb API*.

Jak hodnotíte funkci přidání epizody do kalendáře?

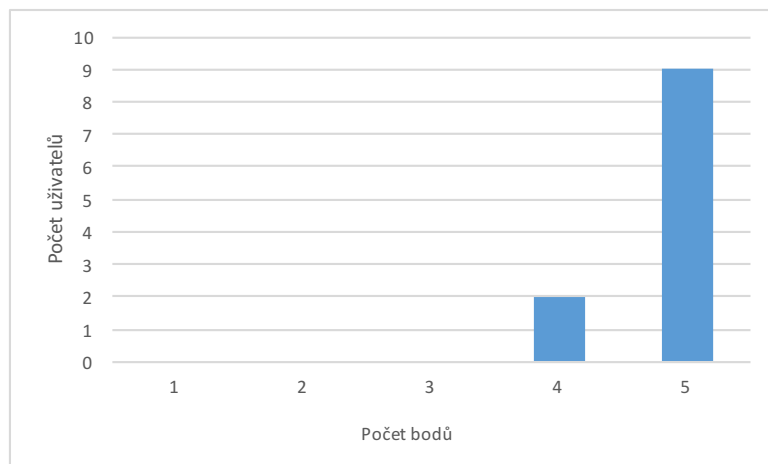
Uživatelé vyhodnotili tuto funkci za velmi užitečnou a ohodnotili nejvyšším počtem bodů. Většina z nich by funkci používala nadále a jen trojice účastníků používali tuto funkci předtím.



Obrázek 5.5: Graf ukazující zájem o kalendář

Jak se vám líbil design aplikace?

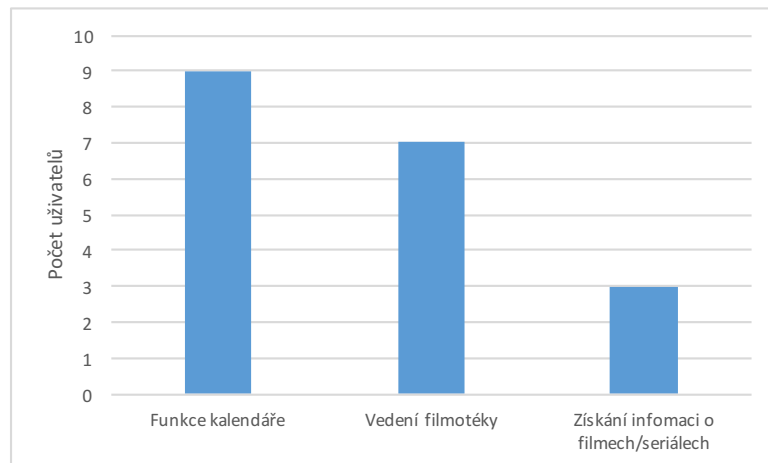
Účastníci testování kladně vyhodnotili design aplikace. V této otázce měli možnost přidat poznámku a přidělit hodnoty na stupnici od 1 do 5, kde 1 je nejhorší a 5 nejlepší možná známka. Jenom jeden uživatel poukázal, že design podstránky “Sezóny” je odlišný od celé aplikace.



Obrázek 5.6: Graf designu aplikace

K čemu byste tuto aplikaci použil?

Na tuto otázku účastníci mohli odpovědět několika možnostmi nebo přidat vlastní. Většina účastníků odpověděla, že tuto aplikaci by používali pro přidávání nových epizod do kalendáře a vedení vlastní filmotéky.



Obrázek 5.7: Graf nejlepších funkcí aplikace

Vyhodnocení testů

Výsledky testování přinesli několik výhod. Byly odstraněny některé chyby na které poukázalo testování a byla provedena analýza dalšího rozvoje aplikace. Na základě zpětné vazby

mužů prohlásit, že se podařilo vytvořit aplikaci, která je pro uživatele logická, pochopitelná a celkem srozumitelná. Podle výsledků testování bylo jasné, že pro každého uživatele má aplikace různé účely. Dojmy uživatelů byly dostatečně pozitivní, lze říci, že i uživatelské rozhraní bylo navrhnuté správně.

Kapitola 6

Závěr

Záměrem této práce bylo vytvořit aplikaci pro vedení filmotéky, kde uživatele sledují filmy a seriály, uchovávají historii zhlednutí, získávají informací o filmech a seriálech, hodnotí je, sdílejí svoje komentáře a dostávají upozornění nových epizodu sledovaných seriálů. Nedílnou součástí práce bylo vytvořit aplikaci s moderním uživatelským rozhraním.

Před samotným návrhem aplikace byly prostudovány různé aplikace s podobným záměrem a způsoby vytváření uživatelských rozhraní pro webové aplikace. Na základě nastudovaných znalostí bylo následně navrženo uživatelské rozhraní. Dále následovalo seznámení s technologiemi využitými při vývoji aplikaci. Následně byl navrhnutý datový model, na základě kterého byla provedena implementace aplikace v jazyce PHP a frameworku Nette. Funkčnost jednotlivých metod je popsána úseky kódů v kapitole o implementaci. Jako poslední část, byl proveden návrh způsobu testování aplikace.

Výsledkem této práce je funkční webová aplikace, v rámci které uživatelé můžou získávat informací o filmech a seriálech. Samozřejmostí je i možnost přidávání filmů a seriálů do vlastního watchlistu. Pomocí aplikace je možné přidávat nové epizody seriálů do kalendáře a tím pádem předcházet zapomínání na nové epizody.

Pomocí testování bylo zjištěno, že aplikace má u uživatelů velký potenciál. Každému uživateli se podařilo vyhledat oblíbený film nebo seriál a přidat ho do watchlistu. Možnost přidání jednotlivých epizod seriálu do svého kalendáře bylo ohodnocené nejvyšším počtem bodů. Taktéž byly ohodnoceny informace o samotných epizodách a možnost vedení vlastní filmotéky. Mužů konstatovat, že stanovené cíle se podařilo splnit.

V budoucnosti by se aplikace mohla rozšířit o další funkcionalitu jako přihlašování a registraci pomocí sociální sítě, například Facebooku, Google nebo Twitter. Další možnosti je ukládání jednotlivých epizod seriálu do watchlistu a taky jejich hodnocení, což může značně zvětšit databázový systém. Poslední částí je připojení Google API pro zobrazení upoutávky seriálu nebo filmu přímo na stránce aplikace.

Literatura

- [1] An open directory of television shows. [Online; navštíveno 05.10.2016].
URL <http://thetvdb.com>
- [2] *GNU General Public License*. [Online; navštíveno 29.09.2016].
URL <http://www.gnu.org/licenses/gpl.html>
- [3] Keep track of your favorite TV shows! - TV Episode Calendar. [Online; navštíveno 01.10.2016].
URL <https://episodecalendar.com>
- [4] Model View Controller. [Online; navštíveno 20.11.2016].
URL <http://ootips.org/mvc-pattern.html>
- [5] Model-view-controller. [Online; navštíveno 20.11.2016].
URL [https://cs.wikipedia.org/wiki/Model-view-controller#/media/File:MVC_Diagram_\(Model-View-Controller\).svg](https://cs.wikipedia.org/wiki/Model-view-controller#/media/File:MVC_Diagram_(Model-View-Controller).svg)
- [6] Movies, TV and Celebrities. [Online; navštíveno 27.09.2016].
URL <http://www.imdb.com>
- [7] Online Mockup, Wireframe, UI Prototyping Tool . [Online; navštíveno 15.11.2016].
URL <https://moqups.com/>
- [8] RESTful Web services: The basics. [Online; navštíveno 14.12.2016].
URL <https://www.ibm.com/developerworks/webservices/library/ws-restful>
- [9] TV API | TVmaze - Add TV information to your website or app. [Online; navštíveno 11.10.2016].
URL <http://www.tvmaze.com/api>
- [10] Your personal TV guide. [Online; navštíveno 27.09.2016].
URL <http://www.tvmaze.com>
- [11] Česko-Slovenská filmová databáze. [Online; navštíveno 04.10.2016].
URL <http://www.csfd.cz>
- [12] MySQL databáze - český manuál. 2002, [Online; navštíveno 1.02.2017].
URL <http://www.junext.net/mysql>
- [13] Planette | Nette Framework. 2008.
URL <https://pln.nette.org>

- [14] Začínáme | Nette Framework. 2008, [Online; navštíveno 14.10.2016].
URL <https://doc.nette.org/cs/2.4/quickstart/getting-started>
- [15] 'Add to Calendar' button for websites and newsletters. 2012, [Online; navštíveno 12.03.2017].
URL <https://www.addevent.com/buttons/add-to-calendar>
- [16] ORM test PHP frameworků - Nette, Prado. 2013, [Online; navštíveno 12.10.2016].
URL <https://www.zdrojak.cz/clanky/orm-test-php-frameworku-nette-prado>
- [17] Castagnetto, J.: *Programujeme PHP profesionálně*. Computer Press, 2002, ISBN 80-7226-310-2.
- [18] Castro, E.; Hyslop, B.: *HTML5 and CSS3: visual quickstart guide*. Computer Press, 2012, ISBN 9788025144657.
- [19] Fritz, B.: OMDb API - The Open Movie Database. [Online; navštíveno 10.10.2016].
URL <http://www.omdbapi.com>
- [20] Strömberg, J.: Search, sort, filters, flexibility to tables, list and more! - List.js. [Online; navštíveno 20.01.2017].
URL <http://listjs.com>

Přílohy

Příloha A

Obsah CD

Obsah jednotlivých složek na CD:

- */src* - Zdrojové kody aplikace
- */doc* - Zdrojové soubory technické zprávy
- *doc.pdf* - Technická zpráva ve formátu PDF
- *README* - Popis spuštění aplikace